

Hepatoprotective effects of systemic ER activation

Single cell analysis

Christian Sommerauer & Carlos Gallardo

11 August, 2022

```
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(Seurat)
library(pagoda2)
library(patchwork)

# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#E98BB6', '#B02262', '#7F9AD7', '#2A2F72', '#7DC7D1', '#339ACD', '#3598B5',
                                c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
colPals$clusters <- setNames(c('#A9D265', '#82506D', '#FA9F1C', '#676A6E'),
                             c('1', '2', '3', '4'))
colPals$celltypes <- setNames(c('#B4272F', '#E5462D', '#FFD1D1', '#F4E54C', '#FBAA3E', '#AA654E', '#B58140',
                                '#B3177E', '#CAC1DD', '#67227D', '#36B449', '#82C349', '#A9D265', '#19944D',
                                c('Cholangiocytes', 'Endothelial cells', 'HsPCs', 'Stromal cells', 'Hepatocytes',
                                'Kupffer cells', 'Monocytes & Monocyte-derived cells', 'T cells', 'NK cells', 'ILCs',
                                'ILC1s', 'B cells', 'cDC1s', 'cDC2s', 'Mig. cDCs', 'pDCs', 'Basophils', 'Eosinophils'))
colPals$inferno <- c('#FCFFA4', '#FCA50A', '#DD513A', '#932667', '#420A68', '#000004')
```

Load data

```
# Liver Cell Atlas data from Guillemin et al. 2022 (https://www.livercellatlas.org/download.php)
liver <- readRDS(file = 'data/livercellatlas-feb2022.rds')

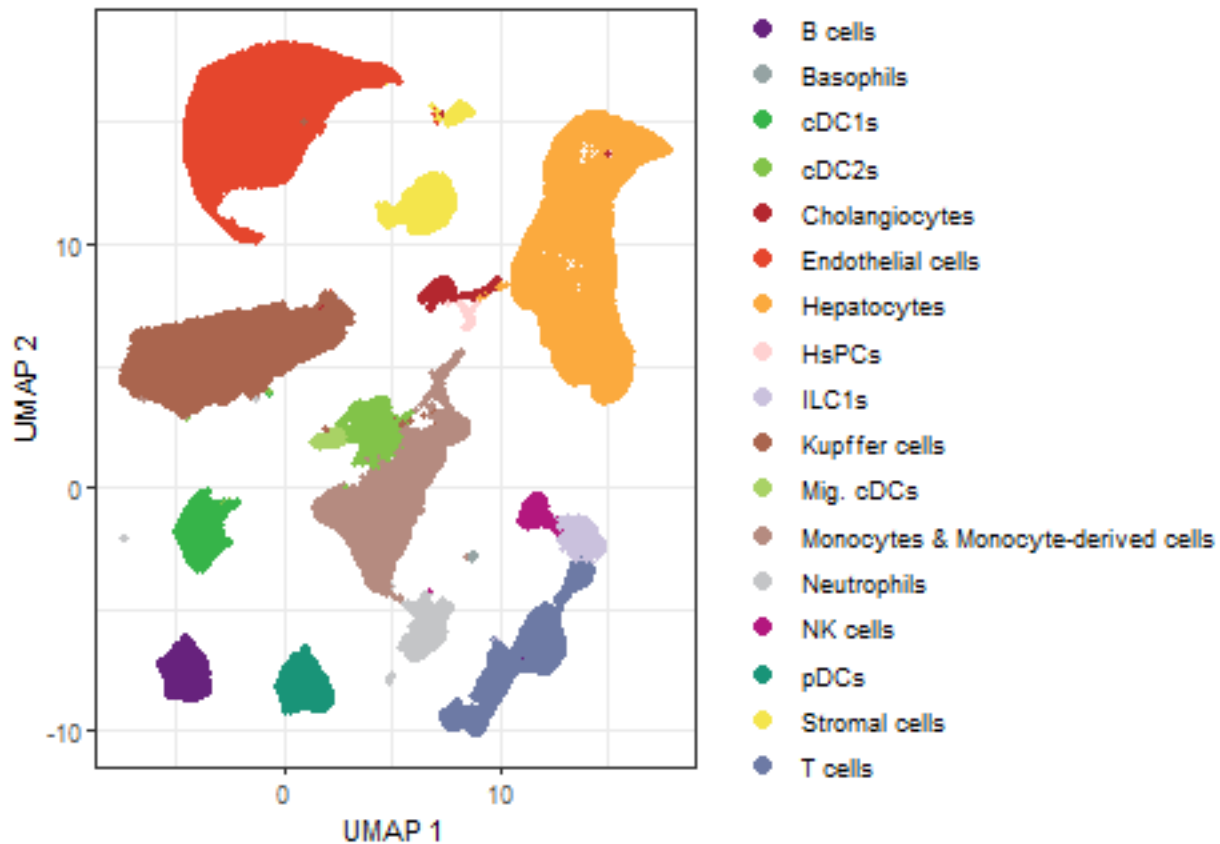
# gene sets
DEG_sets <- readRDS('results/bulkRNAseq_mmus_DEG_sets.rds')
pathway_sets <- readRDS('results/bulkRNAseq_mmus_GSEA_reactome_cluster_sets.rds')
```

Mouse single-cell map

```
df <- liver$mouseStSt$All@meta.data %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells'))

ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=annot, fill=annot, label=annot)) +
```

```
geom_point(shape=21, size=1) +
scale_fill_manual(values = colPals$celltypes) +
scale_color_manual(values = colPals$celltypes) +
xlab("UMAP 1") + ylab("UMAP 2") +
theme_bw() +
guides(colour = guide_legend(override.aes = list(size=3))) +
theme(legend.title = element_blank())
```



Cell type composition

```
df <- liver$mouseStSt$All@meta.data %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells')) %>%
  dplyr::filter(digest == 'inVivo') %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(mouseStSt_n = dplyr::n()) %>%
  dplyr::mutate(mouseStSt_freq = round(mouseStSt_n / sum(mouseStSt_n) * 100, 2))

df2 <- liver$mouseNAFLD$All@meta.data %>%
  dplyr::filter(digest == 'inVivo') %>%
  dplyr::mutate(annot = replace(annot, annot == 'KCs', 'Kupffer cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mono/mono-derived cells', 'Monocytes & Monocyte-derived cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mig cDCs', 'Mig. cDCs')) %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(mouseNAFLD_n = dplyr::n()) %>%
```

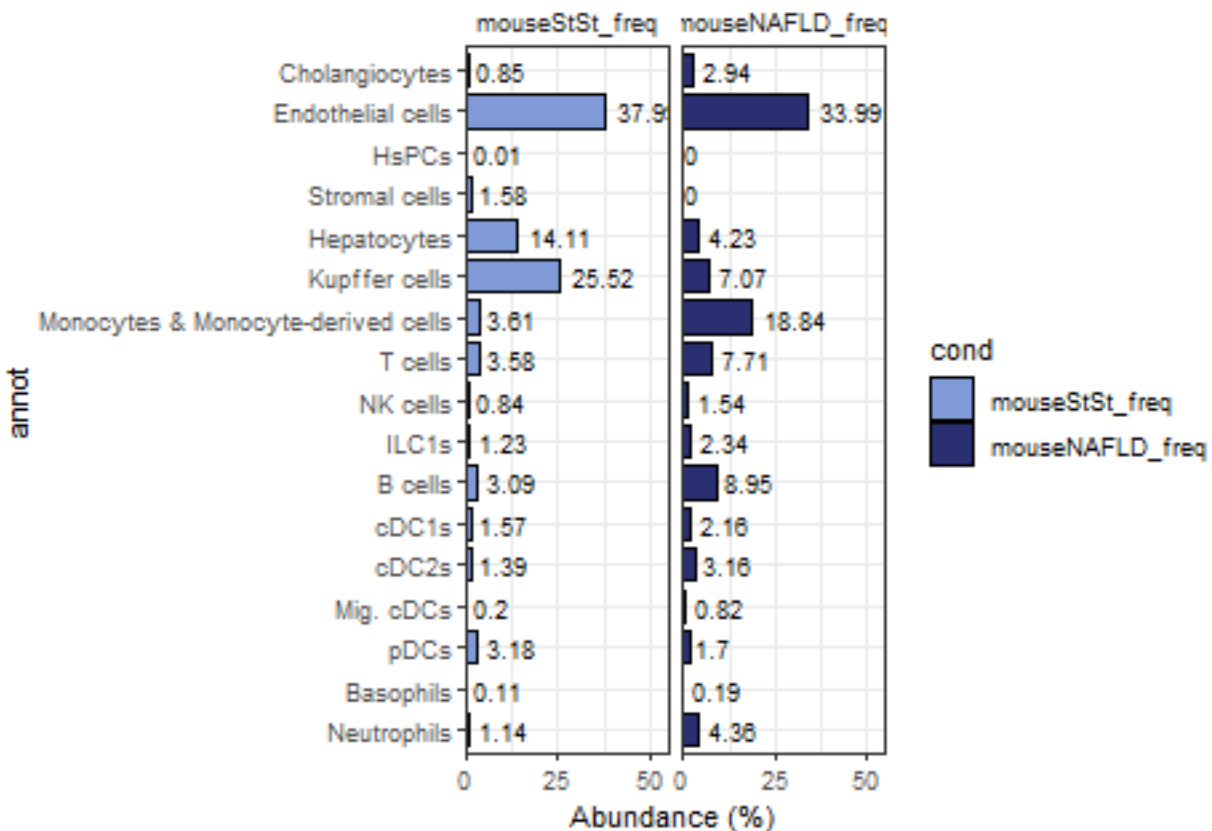
```

dplyr::mutate(mouseNAFLD_freq = round(mouseNAFLD_n / sum(mouseNAFLD_n) * 100, 2))

df <- dplyr::full_join(df, df2, by = 'annot') %>%
  replace(is.na(.), 0) %>%
  tidyr::pivot_longer(cols = c(mouseStSt_freq, mouseNAFLD_freq), names_to = 'cond', values_to = 'freq')
dplyr::mutate(annot = factor(annot, levels = rev(names(colPals$celltypes))),
              cond = factor(cond, levels = c('mouseStSt_freq', 'mouseNAFLD_freq')))

ggplot(df, aes(x=freq, y=annot, fill=cond, label=freq)) +
  geom_bar(color="black", size=0.5, width=0.8, position="stack", stat="identity") +
  geom_text(size=3, hjust=-0.2) +
  scale_fill_manual(values = unname(colPals$conditions[c('CDm', 'HFDm')])) +
  scale_x_continuous(expand = expansion(mult = c(.0, .1)),
                    limit = c(0, 50),
                    breaks = c(0, 25, 50)) +
  facet_wrap(~cond) +
  xlab('Abundance (%)') +
  theme_bw() +
  theme(strip.background = element_blank())

```



Cell type specificity of ER activation signatures

```

# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$mouseStSt$All

```

```

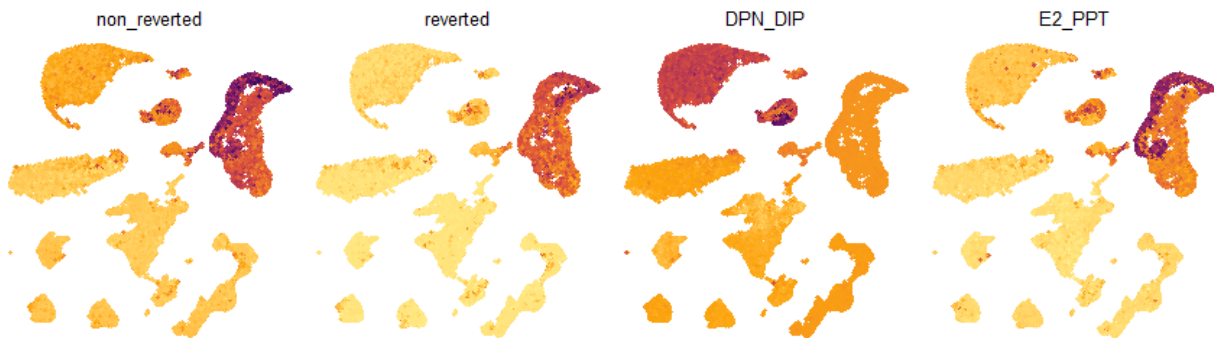
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
  gene.sets = DEG_sets$gene_symbols,
  npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
      UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
        plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')

```



Conservation of cell type specificities in primates

Human

```

# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$human$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,

```

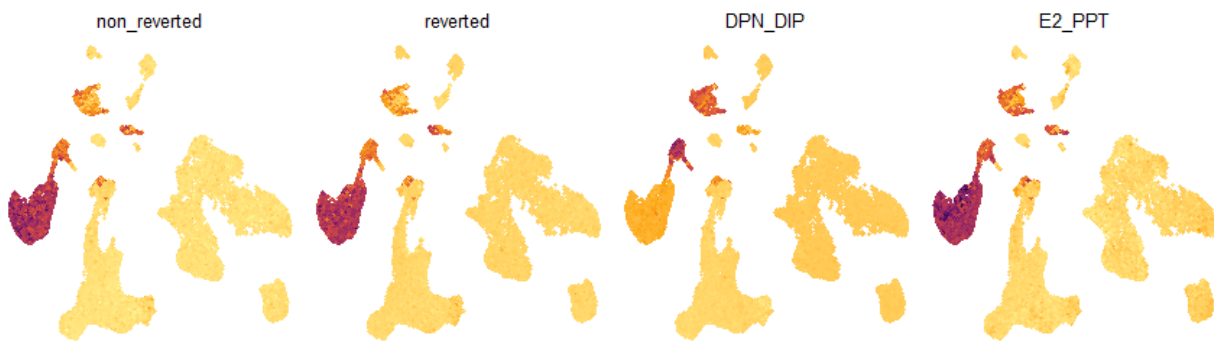
```

      gene.sets = lapply(DEG_sets$gene_symbols, toupper),
      npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')

```



Macaque

```

# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$multisp$monkey
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = lapply(DEG_sets$gene_symbols, toupper),
              npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +

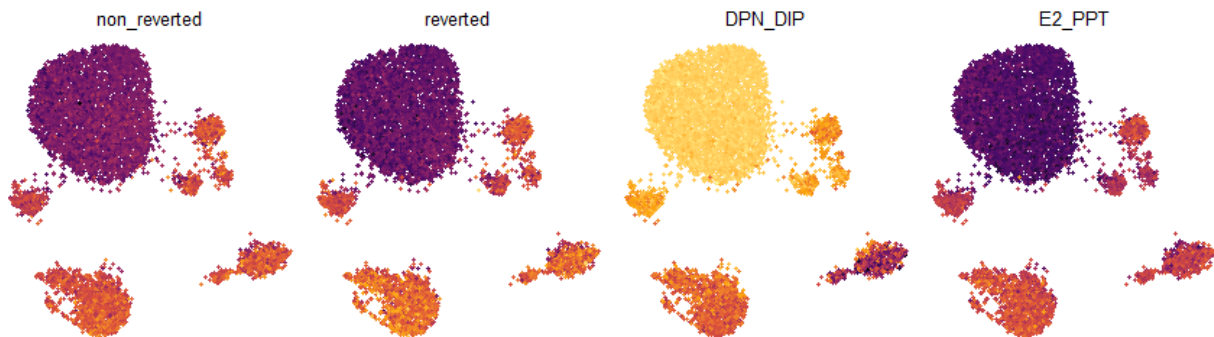
```

```

    scale_color_gradientn(colours = colPals$inferno) +
    theme_void() +
    theme(legend.position = 'none',
          plot.title = element_text(hjust = 0.5)) +
    ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')

```



Liver zonation of ER activation signatures (spatial transcriptomics)

```

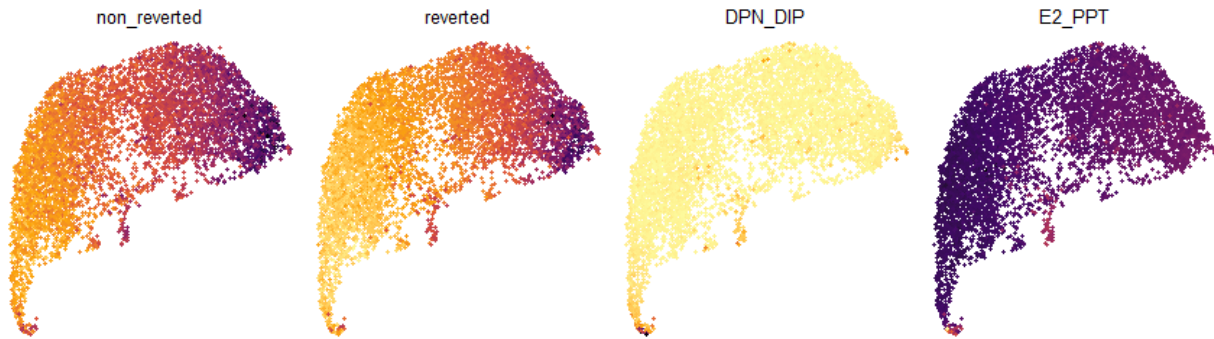
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$mouseStSt$Visium
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(zonationGroup) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = DEG_sets$gene_symbols,
              npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')

```



Pathway enrichment across cell types

Steady State

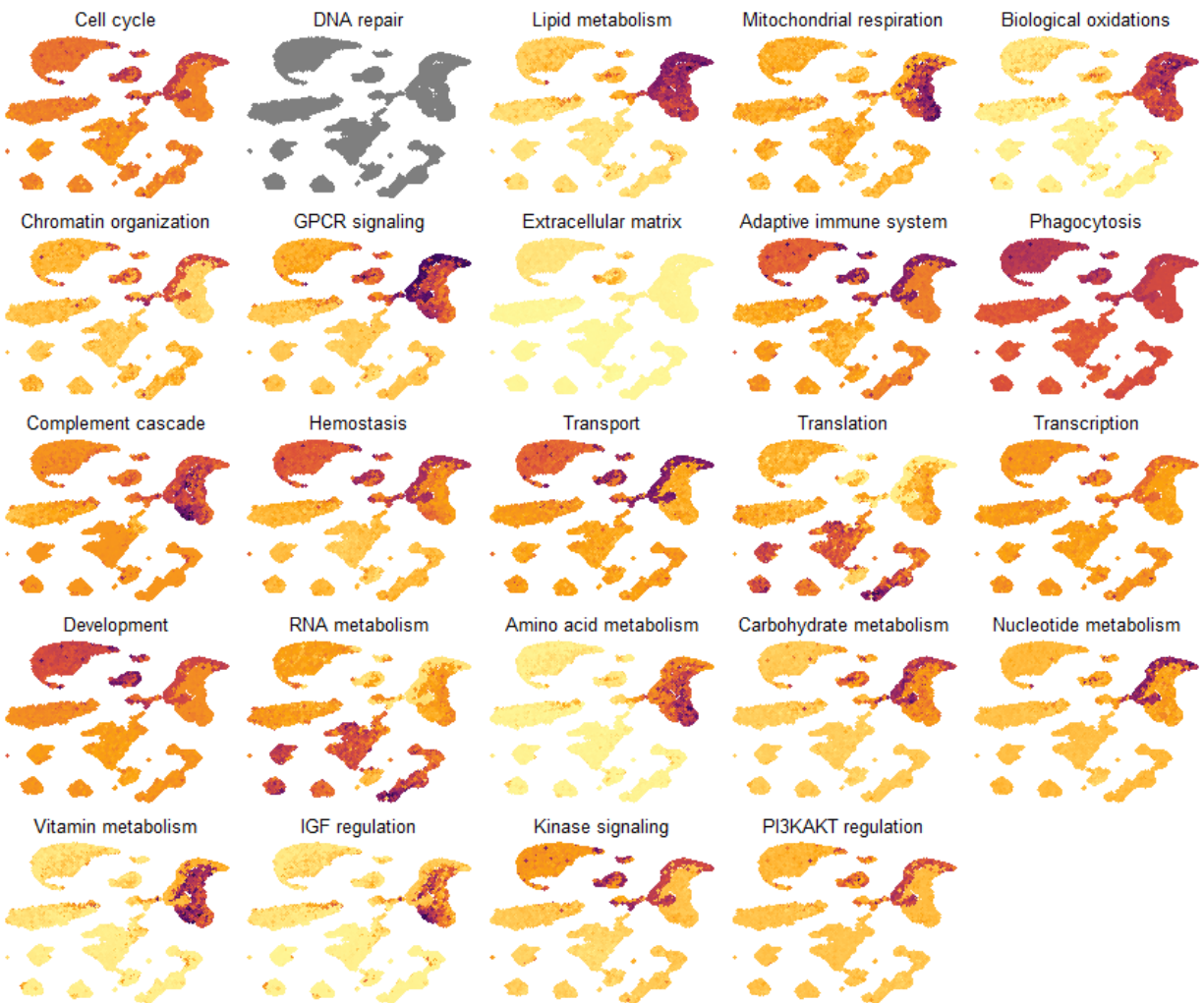
```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$mouseStSt$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
  gene.sets = pathway_sets,
  npcs = 100)

df <- res %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells')) %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, mean)) %>%
  dplyr::mutate(across(`Adaptive immune system`:`Vitamin metabolism`, ~ (. - min(.)) / (max(.) - min(.)))
  tidyr::pivot_longer(cols = `Adaptive immune system`:`Vitamin metabolism`, names_to = 'pathway', values_to = 'score')
  dplyr::mutate(annot = factor(annot, levels = rev(names(colPals$celltypes))),
    pathway = factor(pathway, levels = names(pathway_sets)))

p <- lapply(names(pathway_sets), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
      UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
        plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')
```

NAFLD

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$mouseNAFLD$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
  gene.sets = pathway_sets,
  npcs = 100)

p <- lapply(names(pathway_sets), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
      UMAP_2 = res$UMAP_2)
```

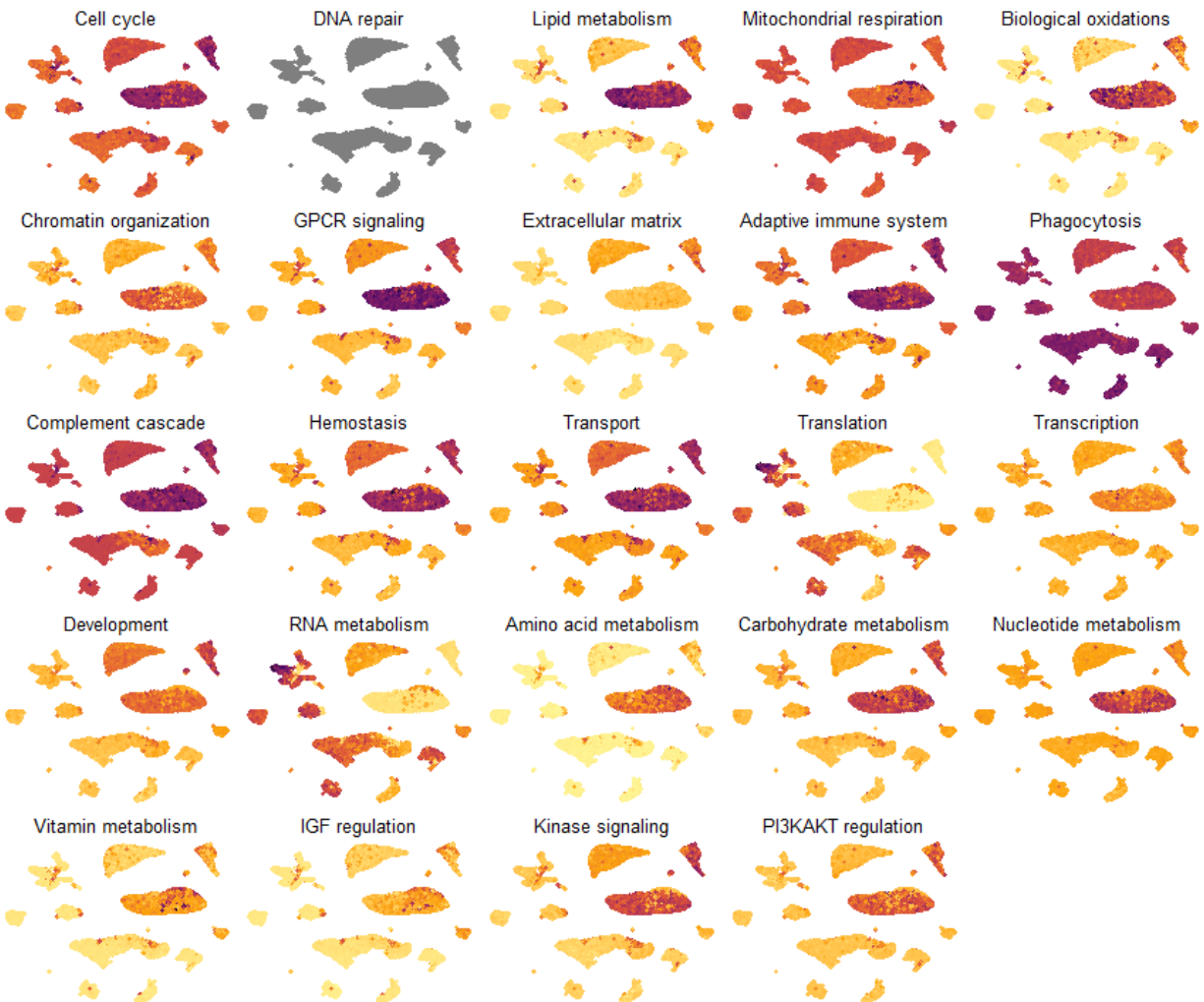


```

df$score <- res[, set]
p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
  geom_point(shape=19, size=1) +
  scale_color_gradientn(colours = colPals$inferno) +
  theme_void() +
  theme(legend.position = 'none',
        plot.title = element_text(hjust = 0.5)) +
  ggtitle(set)
})
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')

```



Comparison of steady state vs NAFLD

```

df2 <- res %>%
  dplyr::mutate(annot = replace(annot, annot == 'KCs', 'Kupffer cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mono/mono-derived cells', 'Monocytes & Monocyte-derived cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mig cDCs', 'Mig. cDCs')) %>%
  dplyr::group_by(annot) %>%

```

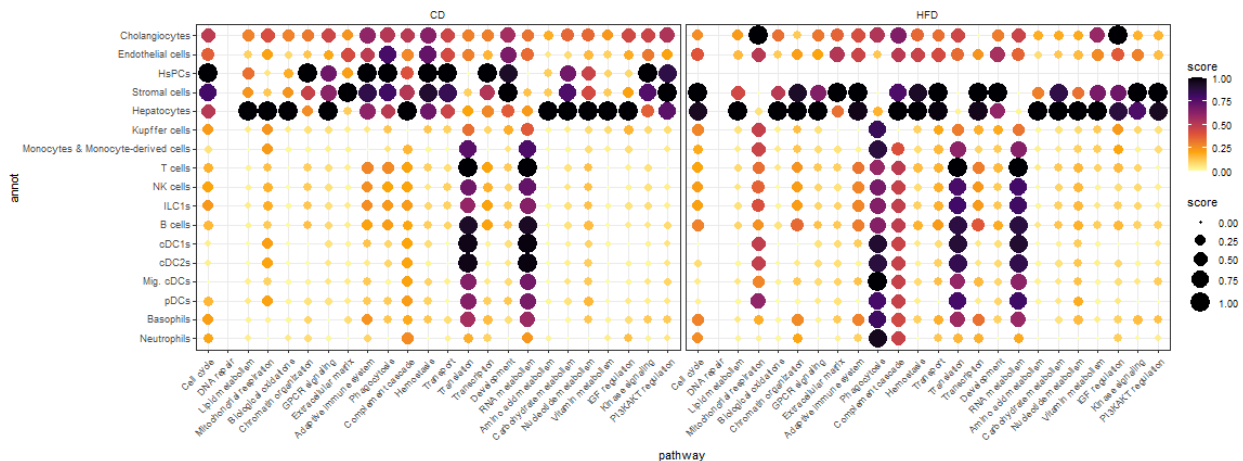
```

dplyr::summarize(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, mean)) %>%
dplyr::mutate(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, ~ (. - min(.)) / (max(.) - min(.)))
tidyr::pivot_longer(cols = `Adaptive immune system`:`Vitamin metabolism`, names_to = 'pathway', values_to = 'score')
dplyr::mutate(cond = 'HFD')

df <- df %>%
dplyr::mutate(cond = 'CD') %>%
dplyr::bind_rows(df2) %>%
dplyr::mutate(cond = factor(cond, levels = c('CD', 'HFD')),
              annot = factor(annot, levels = rev(names(colPals$celltypes))),
              pathway = factor(pathway, levels = names(pathway_sets)))

ggplot(df, aes(x=pathway, y=annot, size=score, color=score)) +
  geom_point() +
  scale_color_gradientn(colours = colPals$inferno) +
  scale_size(range = c(1, 8)) +
  facet_wrap(~cond) +
  theme_bw() +
  theme(strip.background = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text.y = element_text(hjust = 1, vjust = 0.3))

```



Pathway zonation in the liver

```

# subsample single-cell data to 5000 cells per cell type for enrichment analysis
dat <- liver$mouseStSt$Visium
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::group_by(zonationGroup) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = pathway_sets,
              npcs = 100)

p <- lapply(names(pathway_sets), function(set) {

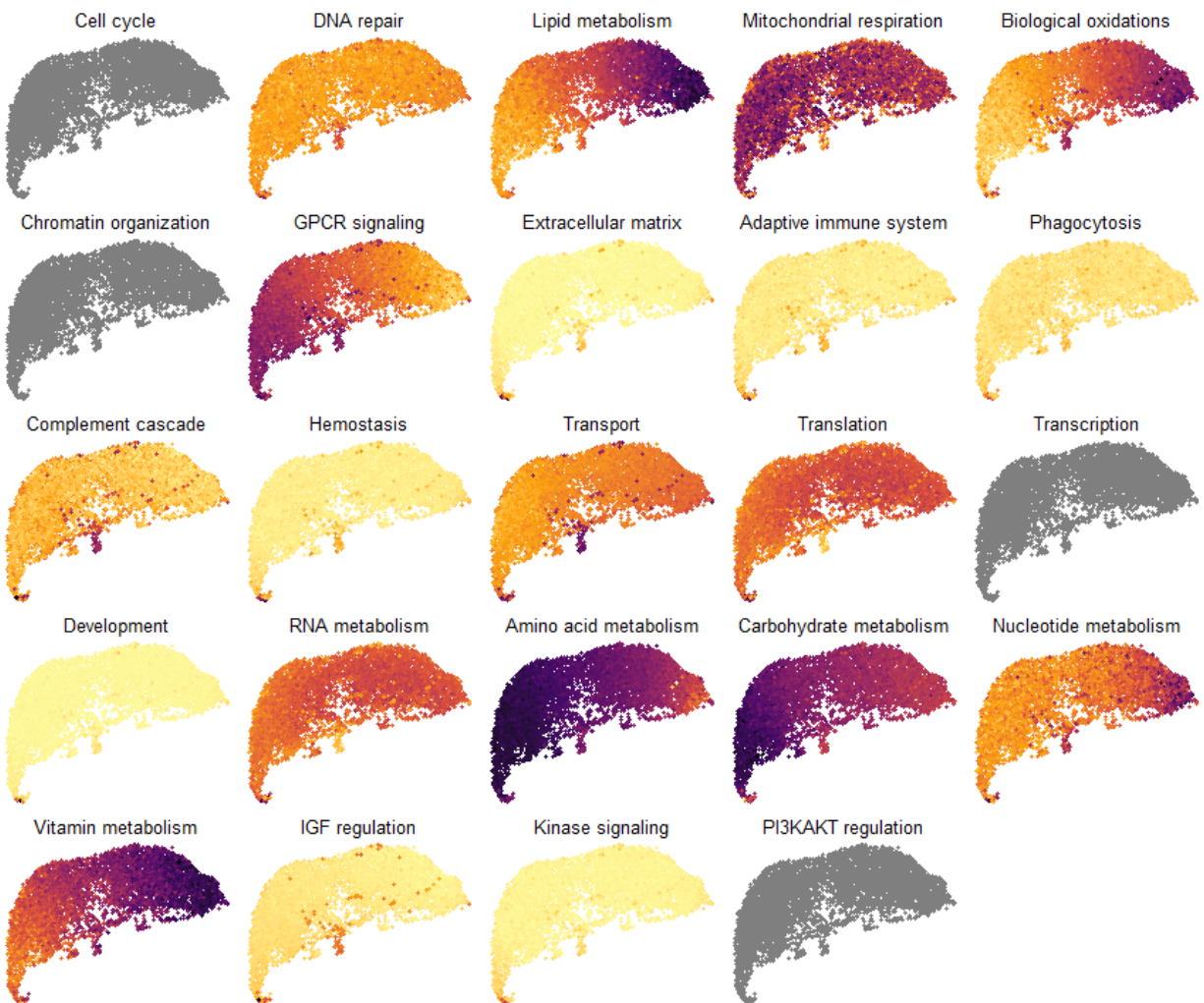
```

```

local({
  df <- data.frame(UMAP_1 = res$UMAP_1,
                  UMAP_2 = res$UMAP_2)
  df$score <- res[, set]
  p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
    geom_point(shape=19, size=1) +
    scale_color_gradientn(colours = colPals$inferno) +
    theme_void() +
    theme(legend.position = 'none',
          plot.title = element_text(hjust = 0.5)) +
    ggtitle(set)
})
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')

```



```

sessionInfo()

## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)

```

```

##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] patchwork_1.1.1    pagoda2_1.0.2      igraph_1.2.6      Matrix_1.3-2
## [5] SeuratObject_4.0.0 Seurat_4.0.2        forcats_0.5.1     stringr_1.4.0
## [9] dplyr_1.0.3        purrr_0.3.4        readr_1.4.0       tidyr_1.2.0
## [13] tibble_3.1.4       ggplot2_3.3.3      tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
## [1] utf8_1.1.4          reticulate_1.18      R.utils_2.10.1
## [4] tidyselect_1.1.0    htmlwidgets_1.5.3    grid_4.0.5
## [7] BiocParallel_1.22.0 Rtsne_0.15           munsell_0.5.0
## [10] codetools_0.2-18    ica_1.0-2            future_1.21.0
## [13] miniUI_0.1.1.1      withr_2.4.1          spatstat.random_2.2-0
## [16] colorspace_2.0-0    Biobase_2.48.0       highr_0.8
## [19] knitr_1.31          rstudioapi_0.13      stats4_4.0.5
## [22] ROCR_1.0-11         tensor_1.5            distillery_1.2
## [25] listenv_0.8.0        labeling_0.4.2        urltools_1.7.3
## [28] polyclip_1.10-0     farver_2.0.3          extRemes_2.1
## [31] parallelly_1.23.0   vctrs_0.3.8          generics_0.1.2
## [34] xfun_0.31           fastcluster_1.1.25    R6_2.5.0
## [37] locfit_1.5-9.4       flexmix_2.3-17        spatstat.utils_2.3-0
## [40] RcppArmadillo_0.10.1.2.2 assertthat_0.2.1     promises_1.1.1
## [43] scales_1.1.1         nnet_7.3-15           gtable_0.3.0
## [46] Cairo_1.5-12.2       globals_0.14.0        conquer_1.0.2
## [49] goftest_1.2-2        drat_0.1.8            rlang_0.4.10
## [52] MatrixModels_0.4-1   splines_4.0.5         lazyeval_0.2.2
## [55] spatstat.geom_2.4-0  broom_0.7.4           brew_1.0-6
## [58] yaml_2.2.1           reshape2_1.4.4        abind_1.4-5
## [61] modelr_0.1.8         backports_1.2.1       httpuv_1.5.5
## [64] tools_4.0.5          sccore_0.1.2          ellipsis_0.3.2
## [67] spatstat.core_2.4-2  RColorBrewer_1.1-2    BiocGenerics_0.34.0
## [70] ggribes_0.5.3        Rcpp_1.0.7            plyr_1.8.6
## [73] dendsort_0.3.3       rpart_4.1-15          deldir_1.0-6
## [76] pbapply_1.4-3        cowplot_1.1.1         scde_2.16.0
## [79] zoo_1.8-8            haven_2.3.1           ggrepel_0.9.1
## [82] cluster_2.1.1        fs_1.5.0              magrittr_2.0.1
## [85] data.table_1.13.6    scattermore_0.7       SparseM_1.78
## [88] p2data_1.0.0         lmtest_0.9-38         triebeard_0.3.0
## [91] reprex_1.0.0         RANN_2.6.1            pcaMethods_1.80.0
## [94] fitdistrplus_1.1-3   matrixStats_0.58.0    hms_1.0.0
## [97] mime_0.9             evaluate_0.14         xtable_1.8-4
## [100] RMTstat_0.3          N2R_0.1.1            readxl_1.3.1

```

## [103] gridExtra_2.3	compiler_4.0.5	KernSmooth_2.23-18
## [106] crayon_1.4.0	R.oo_1.24.0	htmltools_0.5.2
## [109] mgcv_1.8-33	later_1.1.0.1	lubridate_1.7.9.2
## [112] DBI_1.1.1	dbplyr_2.0.0	MASS_7.3-53
## [115] cli_2.3.0	R.methodsS3_1.8.1	parallel_4.0.5
## [118] pkgconfig_2.0.3	plotly_4.9.3	spatstat.sparse_2.0-0
## [121] xml2_1.3.2	rvest_0.3.6	digest_0.6.27
## [124] sctransform_0.3.2	RcppAnnoy_0.0.19	spatstat.data_2.2-0
## [127] rmarkdown_2.14	cellranger_1.1.0	leiden_0.3.7
## [130] Rook_1.1-1	uwot_0.1.11	edgeR_3.30.3
## [133] shiny_1.6.0	Lmoments_1.3-1	quantreg_5.83
## [136] modeltools_0.2-23	rjson_0.2.20	lifecycle_0.2.0
## [139] nlme_3.1-151	jsonlite_1.7.2	viridisLite_0.3.0
## [142] limma_3.44.3	fansi_0.4.2	pillar_1.6.2
## [145] lattice_0.20-41	fastmap_1.1.0	httr_1.4.2
## [148] survival_3.2-7	glue_1.4.2	png_0.1-7
## [151] stringi_1.5.3	irlba_2.3.3	future.apply_1.7.0